SERI/TP-253-1831
UC Category: 62a, b, c, e

The Development of SOLIPH — A Detailed Computer Model of Solar Industrial Process Heat Systems

Charles F. Kutscher

December 1982

To be presented at the ASME
Solar Energy Division 6th Annual
Technical Conference
Orlando, Florida
19-21 April 1983

Prepared Under Task No. 1387.31
WPA No. 351

Solar Energy Research Institute
A Division of Midwest Research Institute

1617 Cole Boulevard
Golden, Colorado 80401

Prepared for the
U.S. Department of Energy
Contract No. EG-77-C-01-4042

THE DEVELOPMENT OF SOLIPH— A DETAILED COMPUTER MODEL OF SOLAR INDUSTRIAL PROCESS HEAT SYSTEMS

Charles F. Kutscher
Solar Energy Research Institute
Golden, Colo. 80401

ABSTRACT

In our effort to generate design tools for solar thermal applications to industrial process heat, we recognized the need to develop a detailed computer model which could simulate hot water and steam systems (unfired boiler and flash tank). Developing our program resulted in a model suited to our specific needs and led to a better understanding of the algorithms used and assumptions made. SOLIPH (for SOLar Industrial Process Heat) is a quasi-steady-state, hour-by-hour model. For each hour of the year, climatological data are read from a TMY (Typical Meteorological Year) weather tape. Outputs are supplied on an hourly, daily, and monthly basis with a cumulative summary for the year.

Before making thousands of SOLIPH runs to generate design tools, SOLIPH was verified against other computer models. In mid-1980 the SSEA working group of ASME proposed an IPH system sample problem to allow comparison of various codes. Results from TRNSYS and DOE-2 (CBS) agreed well. We modeled the same system with SOLIPH and the results showed excellent agreement.

NOMENCLATURE

A	area (m^2)
A_c	collector area (m^2)
C	heat capacitance flow rate (W/K)
C_c	heat capacitance flow rate of cold fluid (W/K)
C_h	minimum heat capacitance of hot fluid (W/K)
C_{min}	minimum heat capacitance flow rate (W/K)
c_p	specific heat (J/kg-K)
c_{pc}	specific heat of collector (J/kg-K)
c_{pw}	specific heat of water (J/kg-K)
D	pipe diameter (m)
f	friction factor
$F_R U_L$	product of collector heat removal factor and loss coefficient (W/m^2-K)
$F_R \tau\alpha$	product of collector heat removal factor, effective transmissivity of glazing(s), and effective absorptivity of absorber
H	head (Pa)
h_g	saturated steam enthalpy (J/kg)
h_o	collector optical efficiency (J/kg)
I	irradiance (W/m^2)
$K_{\alpha\tau}$	incident-angle modifier
L/D	total equivalent pipe
M	collector mass (kg)
$\dot{M}$	collector mass flow rate value (kg/s)
$\dot{M}_L$	load mass flow rate (kg/s)
$\dot{M}c_p$	thermal capacitance (W/K)
$\dot{M}_h$	mass flow rate of hot fluid in heat exchanger (kg/s)
$Q_{ss,loss}$	tank loss
$\dot{Q}_{coll}$	energy rate of collector (W)
$\dot{Q}_{del}$	energy delivered to the load (W)
$\dot{Q}_{inc}$	tank energy increase (W)
$\dot{Q}_{loss}$	heat loss rate (W)
T	temperature (K)
$\bar{T}$	average fluid temperature (K)
T_{amb}	ambient temperature (K)
T_o	storage tank temperature at the beginning of time step (K)
T_s	storage temperature (K)
T_2	return temperature to the collectors (K)
T_∞	steady-state solution for temperature [at $t = \infty$] (K)
t	time (s)
t_{left}	time left after boiling
t'	time step (s)
U	thermal conductance (W/m^2-K)
U_L	heat loss coefficient of collectors (W/m^2-K)
x	distance (m)
ε	input effectiveness
τ_c	collector time constant (s)

INTRODUCTION

Recently SERI published a design handbook for solar industrial process heating systems (1), which is described in another paper by the author elsewhere in these proceedings (2). One purpose of the handbook was to provide designers with simple graphs for determining energy collection for several collector types for both hot water and steam systems. In order to accomplish this, it was first necessary to locate or develop a computer program that could be used to generate the needed data.

A number of available simulation codes were studied and TRNSYS (University of Wisconsin) and SOLTES (Sandia National Laboratories) surfaced as the two best candidates. Further inspection of TRNSYS, however, indicated that new subroutines would need to be written to handle steam systems (both flash-tank and unfired-boiler). SOLTES was set up to handle an unfired boiler, but attempts by two separate programmers over a two-month period to run SOLTES at SERI failed to produce satisfactory results (perhaps due to the large degree of sophistication and versatility intended for the SOLTES code).

Since the numerical methods needed to model the systems we were studying appeared relatively straightforward, we decided to develop our own code. An advantage of doing this is that we would be keenly aware of any assumptions made in the analysis and would have a thorough understanding of the model. In order to quickly obtain a model that was simple to understand and modify and inexpensive to run, we decided not to develop a highly flexible user-oriented code like the others mentioned. The intent was simply to develop an accurate and useful tool for generating the energy prediction graphs.

Before actually writing the code, we designed a flow chart of the first computer model (called SOLIPH for Solar Industrial Process Heat) containing flat-plate collectors, piping, heat exchanger, and storage (see Fig. 1). Once the detailed flow charts were studied by task members, the code was written in Fortran IV and keypunched for use on SERI's CDC 7600 computer.

This paper describes the algorithms used by SOLIPH and the means used to verify SOLIPH results. [Another paper in these proceedings by R. Gee describes the energy prediction design tools developed from SOLIPH runs (3)]. Many of the SOLIPH algorithms are similar to those used by other codes, but others, such as the determination of overnight collector losses and the unfired-boiler and flash-tank models, are unique to SOLIPH. The discussion herein follows a description of the SOLIPH computer program contained in an appendix of the SERI IPH handbook.

Like other solar simulation codes (e.g., TRNSYS, SOLTES, etc.), SOLIPH is a quasi-steady-state, hour-by-hour model. For each hour of the year, climatological data (time, direct normal insolation, total horizontal insolation, and ambient temperature) are read from a TMY (Typical Meteorological Year) weather tape, available for 26 different weather stations. The executive routine reads all the system and climatological input data and then calls various subroutines (collector, pipe, heat exchanger, storage, etc.) around closed piping loops. A starting-point temperature is chosen for each pipe loop (the collector array inlet temperature in the collector loop), and energy balances are performed on each component. The program cycles through each loop as many times as necessary until the temperature distribution is essentially unchanged from the previous iteration (to within a specified temperature convergence criterion). Each hour, new climatological and load inputs are used, and the program arrives at a new steady-state solution, until the entire year has been modeled.

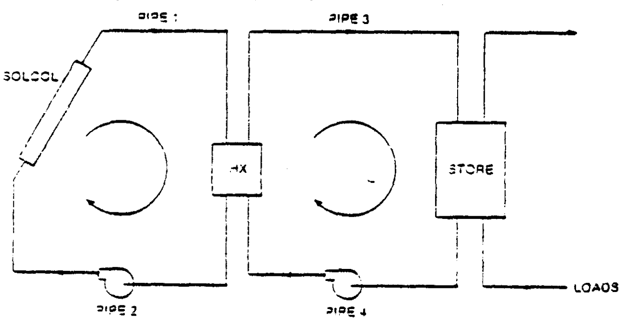

Fig. 1 A Simple SOLIPH Configuration

Figure 1 shows the original SOLIPH model. The starting point for this configuration is the inlet collector temperature. The executive routine calls the collector subroutine, SOLCOL, which, based on input collector parameters and climatological data, supplies a collector array outlet temperature. This is used as the inlet temperature for pipe 1. The pipe subroutine, PIPE, is called and, based on input pipe insulation and ambient temperature, supplies an outlet temperature. The outlet temperature is used as the hot-side inlet temperature to the heat exchanger. The heat exchanger routine, HX, uses input effectiveness and the hot and cold inlet temperatures to compute the two outlet temperatures. The hot outlet temperature is then used as input for pipe 2. The PIPE subroutine is again called, and its outlet value replaces the inlet collector temperature we started with. If these two temperatures are not sufficiently close, the loop must be repeated.

First, however, the second loop is computed. The cold-side outlet temperature of the heat exchanger is used as input for pipe 3. The outlet from PIPE is used as the inlet storage temperature. The LOADS subroutine is called to determine the load at that hour and to allow a complete energy balance on the storage tank, and then STORE is called to determine a new storage tank temperature. This temperature is used as the input for pipe 4, and the output temperature from PIPE is compared with the inlet cold-side heat-exchanger temperature originally assumed, thus completing the second loop. If either loop has not converged within the specified tolerance, both are reiterated.

To make the code as easy to understand as possible, energy calculations are done in the subroutines where they are most appropriate rather than at the end of the executive routine. So, for example, pipe energy loss is calculated in the PIPE subroutine, storage loss in STORE, etc. Since only temperature values are needed for loop iteration, calculating all energy values during each cycle would waste computer time. SOLIPH is set up to do these calculations only on the last iteration.

Output temperatures and energy values are supplied on hourly, daily, and monthly bases with an annual cumulative summary. Energy collected by the array, energy delivered to the load, and pumping energy are outputs. Energy losses are given separately for piping and storage and are also broken down according to whether they are operational (during collector pump operation) or nonoperational losses. Collector array efficiencies are given relative to useful energy in the plane of the collectors as well as to total horizontal energy. System efficiencies are supplied as the ratio of energy delivered (to the load) to insolation. Table 1 is a sample monthly summary.

Table 1 Sample Monthly Summary Output

Day	ITH[a] (GJ)	IBN (GJ)	IAVAIL (GJ)	IAPER (GJ)	QCOLL (GJ)	ETA(1) (%)	ETA(3) (%)	QDEL (GJ)	OLOSS (GJ)	NOLOSS (GJ)	SLOSS (GJ)	ETA(2) (%)	EPAR (GJ)
1	16.05	15.07	20.18	19.38	0.00	0.0	0.0	0.96	0.000	0.283	2.790	4.8	0.00
2	20.60	24.84	28.88	28.69	5.33	25.9	18.6	0.00	0.093	0.518	2.586	0.0	0.27
3	23.52	36.29	35.40	34.00	6.97	29.6	20.5	4.05	0.103	0.541	2.604	11.4	0.27
4	25.44	46.97	41.68	40.52	12.15	47.8	30.0	6.52	0.137	1.627	2.569	15.7	0.37
5	32.45	74.89	58.58	55.78	17.52	54.0	31.4	11.94	0.166	0.864	2.913	20.4	0.37
6	32.66	69.16	56.83	54.57	17.14	52.5	31.4	12.95	0.165	0.839	2.958	22.8	0.37
7	29.67	60.35	50.57	48.56	14.87	50.1	30.6	11.61	0.147	0.803	2.685	23.0	0.37
8	32.51	69.76	56.97	54.55	16.20	49.8	29.7	12.04	0.171	1.054	3.034	21.1	0.37
9	29.38	54.53	48.85	47.94	12.44	42.3	25.9	9.40	0.171	1.033	3.083	19.2	0.37
10	17.98	20.80	24.64	24.25	4.01	22.3	16.5	0.00	0.060	0.389	2.725	0.0	0.16
11	10.74	7.47	12.54	11.88	0.00	0.0	0.0	0.00	0.000	0.505	2.948	0.0	0.00
12	28.63	52.03	46.82	45.28	12.25	42.8	27.0	5.06	0.162	1.608	2.979	10.8	0.37
13	26.32	43.74	42.21	41.30	8.98	33.5	21.7	5.66	0.148	0.738	3.096	13.4	0.32
14	31.17	65.71	54.46	52.18	14.59	46.8	28.0	8.52	0.172	1.029	3.066	15.6	0.37
15	23.90	44.14	39.56	38.29	8.82	36.9	23.0	6.63	0.121	0.746	3.070	16.8	0.27
16	29.62	66.41	53.57	51.65	14.38	48.6	27.8	8.15	0.177	1.319	3.058	15.2	0.37
17	30.25	45.29	45.83	44.66	9.39	31.0	21.0	5.85	0.171	1.218	3.200	12.8	0.37
18	29.97	66.34	53.70	51.33	14.24	47.5	27.7	8.26	0.173	1.287	3.133	15.4	0.37
19	30.86	72.15	56.65	53.83	15.61	50.6	29.0	10.61	0.173	1.009	3.124	18.7	0.37
20	31.00	72.73	56.98	54.12	16.12	52.0	29.8	11.50	0.170	1.042	3.075	20.2	0.37
21	30.94	70.22	56.02	53.39	15.78	51.0	29.6	11.54	0.169	1.048	3.068	20.6	0.37
22	30.59	65.98	54.11	51.76	14.37	47.0	27.3	10.43	0.171	1.078	3.113	19.4	0.37
23	30.72	68.99	55.35	52.82	15.29	49.8	29.0	10.53	0.170	0.985	3.076	19.0	0.37
24	31.37	72.75	57.39	54.72	16.63	53.0	30.4	11.82	0.170	1.043	3.079	20.6	0.37
25	24.32	44.36	40.74	39.41	10.31	41.5	26.2	8.36	0.117	1.005	2.958	20.5	0.27
26	28.06	50.96	46.09	44.73	12.86	45.8	28.7	3.15	0.138	1.049	2.849	17.7	0.32
27	12.45	5.08	13.08	12.41	0.00	0.0	0.0	0.00	0.000	0.582	3.048	0.0	0.00
28	30.51	72.40	56.32	53.52	14.04	46.0	25.2	6.33	0.183	1.064	3.296	11.2	0.37
29	29.46	66.30	52.93	50.41	11.57	39.3	23.0	6.56	0.184	1.157	3.411	12.4	0.37
30	30.58	73.21	56.59	53.72	14.03	45.9	26.1	8.81	0.184	1.099	3.368	15.6	0.37
31	23.18	34.25	34.25	32.78	3.38	14.6	10.3	0.00	0.096	0.751	3.178	0.0	0.21
b	835.9	1633.7	1407.7	1352.4	349.3	41.8	25.8	222.3	4.26	29.31	93.14	16.0	9.57

[a]ITH = Total daily horizontal irradiation times collector array aperture area (GJ).
IBN = Daily direct normal irradiation times collector aperture area (GJ).
IAVAIL = Daily usable irradiation in collector plane times collector aperture area (GJ).
IAPER = Daily irradiation in collector plane times collector aperture area (GJ).
QCOLL = Daily energy collected by collector array (GJ).
ETA(1) = Collector array efficiency—QCOLL/ITH.
ETA(3) = Collector array efficiency—QCOLL/IAVAIL.
QDEL = Daily collected energy delivered to the load (GJ).
OLOSS = Daily operational thermal system losses (during pump operation) (GJ).
NOLOSS = Daily nonoperational system thermal losses (during pump shutdown) (GJ).
SLOSS = Daily thermal losses from storage tank (GJ).
ETA(2) = System thermal efficiency—QDEL/IAVAIL.
EPAR = Daily parasitic (pumping) energy (GJ).

[b]Monthly total or average.

Before we describe the various subroutines, we must point out again that SOLIPH is not a highly user-oriented code like TRNSYS or SOLTES. To change the system configuration, the user does not change input data as he or she would for one of the other codes. Rather, the user changes the Fortran programming statements in the executive program. Thus, a CALL STORE might be replaced by a CALL HX. Although this is not an elegant approach, it is not complex and results in shorter run times. The following section describes the basic algorithms used in each SOLIPH subroutine.

THE SOLIPH SUBROUTINES

Subroutine SOLCOL

This subroutine models a stationary solar collector array. Direct normal and total horizontal radiation and ambient temperature from the weather tape are used. Values of $F_R\tau\alpha$, F_RU_L, incident-angle modifier coefficients, and the collector time constant are input, as well as collector tilt, azimuth, and ground-cover ratio.

First, insolation in the collector plane is calculated. SOLIPH does this in the following steps:

- calculates hour-angle and declination (a day-of-the-year function is used to determine declination),
- calculates incident angle and converts normal beam radiation to collector plane,
- calculates diffuse radiation in horizontal plane,
- converts diffuse radiation to collector plane (including ground and sky terms), and
- adds direct, diffuse, and reflected beam radiation in plane.

Now, it must be determined whether the collector pump is on. If the pump has been operated, this determination is based on the difference between previous collector outlet temperature and storage. If the pump is not on, as in early morning, a stagnation temperature is calculated as follows:

$$T_s = T_{amb} + \frac{K_{\tau\alpha}\eta_o I}{F_R U_L} , \qquad (1)$$

where η_o is the optical efficiency. The stagnation temperature is compared to the storage tank temperature, and if it is sufficiently higher, the pump is turned on. Several flags provide a dead-band temperature range that allows for a typical $\Delta T_{on}/\Delta T_{off}$ control scheme. If the pump is on, collector efficiency is calculated from the Hottel-Whillier-Bliss equation (incident-angle corrected) and multiplied by insolation and collector area to yield energy collected. Collector outlet temperature is then

$$T_{out} = T_{in} + \frac{\dot{Q}_{coll}}{\dot{M}c_p} \quad . \tag{2}$$

If the pump is off, cool-down loss from the collectors is determined. For collector mass M, specific heat c_{pc}, and loss coefficient U, we solve a time-dependent, first-order ordinary differential equation:

O.D.E.: $$Mc_{pc}\frac{dT}{dt} = -U_L A_C(T - T_{amb}) + K_{\alpha\tau}\eta_o IA$$

I.C.: $$T|_{t=0} = T_o$$

Soln.: $$T = T_{amb} + \frac{K_{\alpha\tau}\eta_o I}{U_L} + \left(T_o - T_{amb} - \frac{K_{\alpha\tau}\eta_o I}{U_L}\right)e^{-\frac{U_L A_C}{Mc_{pc}}t} \quad . \tag{3}$$

$K_{\alpha\tau}\eta_o I/U_L$ in the differential equation accounts for stagnation heating. The MC value for the collectors is found from the collector time constant τ_c as determined by ASHRAE 93-77 and input as

$$\tau \equiv \frac{Mc_{pc}}{2\dot{M}c_p} \quad , \tag{4}$$

$$Mc_{pc} = 2\tau_c\dot{M}c_p \quad , \tag{5}$$

where $\dot{M}$ is the collector mass flow-rate value used in the ASHRAE test. Once the array temperature is calculated, energy loss is determined as Mc_{pc} times the difference between that temperature and the one for the previous time step. (At shutdown, the average of inlet and outlet temperatures is used for the first value.) These calculations apply only to the collectors and not the header pipes. The headers are lumped into the pipe runs to and from the collector array, and their losses are determined with the supply/return pipe losses.

Subroutine TROUGH

The parabolic trough subroutine first calculates incident angle for a north-south or east-west trough array. Direct insolation on the aperture is then calculated, correcting for row-to-row shading losses. Pump status is determined by comparing available radiation to an input critical intensity. If the radiation is sufficient, an incident-angle modifier is calculated from input coefficients. The average fluid temperature $\overline{T}$ is determined as follows: the energy collected per unit area is the product of efficiency and irradiation and also the product of mass flow rate, specific heat, and fluid temperature rise. Setting these quantities equal, we have

$$\eta_o A_C I - U_1(\overline{T} - T_{amb}) - U_2(\overline{T}_{amb})^2 = 2\dot{M}c_p(\overline{T} - T_{in}) \quad . \tag{6}$$

SOLIPH then uses the quadratic formula to solve this equation for $\overline{T}$. When the average fluid temperature is known, efficiency can be calculated from the input coefficients. (Second-order equations are used for the troughs.) The energy collected is then $\eta_c IA_c$, and outlet temperature is calculated just as it is for stationary collectors.

For situations in which the pump is off, the decaying receiver temperature and associated energy loss are calculated just as for stationary collectors. The exception is that an input receiver mass is used, so that it does not have to be calculated from an input time constant. The stagnation term is not included because parabolic trough collectors would be defocused when the pump is off.

Subroutine PIPE

If the pump is operating, pipe outlet temperature is determined from inlet temperature, ambient temperature, and input insulation, using the solution of a first-order ordinary differential equation (see Fig. 2):

O.D.E: $$-\dot{M}c_p\frac{dT}{dx} = U\pi D(T - T_{amb})$$

B.C.: $$T|_{x=0} = T_{in}$$

Soln.: $$T = T_{amb} + (T_{in} - T_{amb})e^{-\frac{U\pi D}{\dot{M}c_p}x} \quad . \tag{7}$$

Energy lost is then

$$\dot{Q}_{loss} = \dot{M}c_p(T_{out} - T_{in}) \quad . \tag{8}$$

To determine parasitic power, the Reynolds number is calculated first to determine which flow regime is present. The Colebrook equation determines the friction factor when flow is turbulent, and f is calculated as 64/Re when flow is laminar. For transitional flow, a simple linear interpolation between the two is used. Values for total equivalent pipe L/D, including fittings and elbows, are user inputs. Head is then determined as $H = f(L/D)(v^2/2g)$. For simplicity, user inputs for average collector and heat-exchanger pressure drops are added here, although they could also be in separate subroutines.

If the pump is off, the solution for temperature decay is similar to that for the collectors:

$$T = T_{amb} + (T_o - T_{amb})e^{-\frac{UA}{Mc_p}t} \quad . \tag{9}$$

The Mc_p product includes both the fluid and pipe metal. (Insulation heat capacity is usually neglected.) The cool-down energy loss is then the

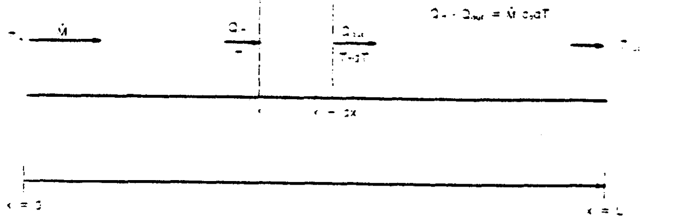

Fig. 2 Steady-State Pipe Loss Energy Balance

product of Mc_p and the average temperature change over a time step. Note that this can be negative for the collectors and pipe. If they have cooled down to ambient and the ambient temperature rises, there will be a heat gain.

Subroutine HX

The heat-exchanger subroutine uses the flow rates, hot and cold inlet temperatures, and the input effectiveness ε to compute the hot and cold outlet temperatures using simple heat-exchanger effectiveness equations (see Fig. 3). Thus, we have

$$T_{c\text{-}out} = T_{h\text{-}in} - \varepsilon \frac{C_{min}}{C_h} (T_{h\text{-}in} - T_{c\text{-}in}) \qquad (10)$$

and

$$T_{h\text{-}out} = T_{c\text{-}in} + \varepsilon \frac{C_{min}}{C_c} (T_{h\text{-}in} - T_{c\text{-}in}) \quad , \qquad (11)$$

where C refers to heat-capacitance flow rate, i.e., $\dot{M}c_p$. Heat-exchanger thermal losses are considered to be negligible.

Subroutine LOADS

The LOADS subroutine supplies the load flow rate and return temperature at each hour needed to perform the loop energy balances. Load profiles and load delivery control schemes are changed by changing the code in LOADS. In a typical setup, LOADS first checks to see if a load exists for a given hour. If so, it calculates the difference between the storage tank and load return temperatures to determine whether flow should occur through storage. Several flags allow for a dead band, so that a $\Delta T_{on}/\Delta T_{off}$ control strategy can be used. The LOADS subroutine also has a mix valve capability that allows only a fraction of the load return fluid to pass through storage and then be mixed with bypassed load return fluid to limit the load supply temperature to a set value.

Subroutine STORE

The basic storage subroutine assumes mixed storage and takes skin losses into account. Flow into and out of storage occurs both on the collector side and the load side. If we call the inlet temperature on the collector (hot) side T_h, the load return (cold) temperature to the tank T_c, and the mixed storage temperature T (same as return to collectors and load supply) (see Fig. 4), we have

$$\text{O.D.E.:} \quad Mc_p \frac{dT}{dt} = \dot{M}_h c_p (T_h - T) - \dot{M}_c c_p (T - T_c) - UA(T - T_{amb})$$

$$\text{I.C.:} \quad T\big|_{t=0} = T_0$$

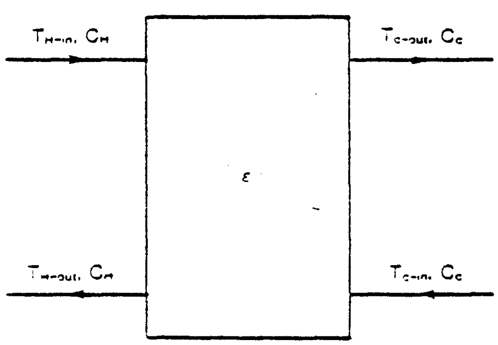

Fig. 3 SOLIPH Heat-Exchanger Model

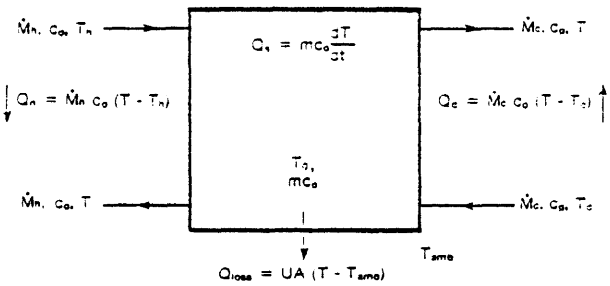

Fig. 4 SOLIPH Storage Tank Heat Balance

$$\text{Soln.:} \quad T = T_\infty + (T_0 - T_\infty)e^{-\frac{\dot{M}_h c_p + \dot{M}_c c_p + UA}{Mc_p} t} \quad , \qquad (12)$$

where

$$T_\infty = \frac{\dot{M}_h c_p T_h + \dot{M}_c c_p T_c + UAT_{amb}}{\dot{M}_h c_p + \dot{M}_c c_p + UA} \quad . \qquad (13)$$

Note that, in the case of no flow on either side, the solution reduces to the same form as that for the pipe.

If T_0 is the storage tank temperature at the beginning of the time step, substituting in the time step (one hour) for t in the above solution will give the tank temperature at the end of the time step. For the energy balance to work, however, we must use the average tank temperature during the time step t'; i.e., we must integrate the above solution and divide by the time step. Thus,

$$\overline{T} = \frac{1}{t'} \int_0^{t'} T \, dt \quad . \qquad (14)$$

The solution is

$$\overline{T} = T_\infty - \frac{T_0 - T_\infty}{t'} \cdot \frac{Mc_p}{X} \left(e^{-\frac{X}{Mc_p} t'} - 1 \right) \quad , \qquad (15)$$

where $X = \dot{M}_h c_p + \dot{M}_c c_p + UA$. The energy lost from storage is determined by integrating the product of UA and the difference between instantaneous storage tank temperature and ambient temperature. Thus,

$$Q_{loss} = \int_0^{t'} UA(T - T_{amb}) \, dt \quad , \qquad (16)$$

the solution of which is

$$Q_{loss} = UA(T_\infty - T_{amb}) t' + UA \frac{Mc_p}{X} (T_0 - T_\infty) \left(1 - e^{-\frac{X}{Mc_p} t'} \right) . \qquad (17)$$

(We should emphasize here that, in all cases, SOLIPH merely uses the solution to a differential equation or integral rather than solving it.) The energy delivered to the load is calculated as the product of the load mass flow rate, specific heat, and difference between load return temperature and average storage temperature:

$$\dot{Q}_{del} = \dot{M}_c c_p (\overline{T} - T_c) \quad . \qquad (18)$$

Subroutine TCSTORE

The thermocline storage subroutine (TCSTORE) is a simple 2-node model that works by stacking two mixed storage tanks, one on top of the other, and calling STORE twice. The storage volume and UA are divided equally between the two tanks. Energy delivered is the product of load flow rate, specific heat, and the difference between the top tank temperature and the load return temperature. Total energy loss is the sum of the losses from each tank.

Subroutine BOILER

This subroutine models an unfired boiler. The boiler is divided into a preheat section and a boiling section; total area is held constant. The temperature of the saturated steam to be delivered is specified, as are inlet hot (collector) and cold (load) temperatures and overall U value. Because thermal resistance on the oil (hot) side is dominant in both sections, using one U value for both is a reasonable approximation. Oil flow can be varied to yield a constant steam flow (up to a specified limit), or a constant hot flow can yield a variable steam output. Outputs of the subroutine are hot-side outlet temperature and the unknown flow.

The algorithm solves four simultaneous equations. Two are heat-exchanger equations (using $Q = UA \cdot LMTD$), one for the preheater, the other for the boiler. The other two are energy balance equations, one across the preheater, the other for the boiler (see Fig. 5).

Boiler:

$$(\dot M c_p)_{oil}(T_h - T_i) = \frac{UA_1(T_h - T_i)}{\ln\left(\dfrac{T_h - T_s}{T_i - T_s}\right)} \qquad (19)$$

$$(\dot M c_p)_{oil}(T_h - T_i) = \dot M_w h_{fg} \quad . \qquad (20)$$

Preheater:

$$(\dot M c_p)_{oil}(T_i - T_c) = \frac{UA_2[(T_i - T_s) - (T_s - T_w)]}{\ln\left(\dfrac{T_i - T_s}{T_c - T_w}\right)} \qquad (21)$$

$$(\dot M c_p)_{oil}(T_i - T_c) = \dot M_w c_{pw}(T_s - T_w) \quad . \qquad (22)$$

This gives us four equations in four unknowns: T_c, T_i, A_1 or A_2, and the unspecified flow. (We know the total area, $A_1 + A_2$, but do not know the breakdown.) Since the equations are nonlinear (two contain log terms), we cannot solve them by Gaussian elimination. We must combine them into one equation in R, defined as $(\dot M c_p)_{oil}/(\dot M c_p)_w$ to obtain

$$U(A_1 + A_2) = (\dot M c_p)_{oil} R \left\{ \ln\left[\frac{T_h - T_s}{T_h - T_s - \frac{1}{R}\left(\frac{h_{fg}}{c_{pw}}\right)}\right] + \frac{1}{1-R}\ln\left[\frac{T_h - T_s - \frac{1}{R}\left(\frac{h_{fg}}{c_{pw}}\right)}{T_h - T_w - \frac{1}{R}\left(\frac{h_{fg}}{c_p} + T_s - T_w\right)}\right]\right\} . \qquad (23)$$

Thus, we have one equation in one unknown, R. Subtracting $U(A_1 + A_2)$ from both sides, we have an equation of the form $f(R) = 0$. SOLIPH solves for the R root by the Newton-Raphson method. Typically, fewer than five iterations are required. Once R, and hence the unknown flow rate, is determined, the energy delivered is calculated as

$$\dot Q_{del} = \dot M_w [h_{fg} + c_{pw}(T_s - T_w)] \quad , \qquad (24)$$

and the oil-return temperature as

$$T_c = T_h - \frac{Q_{del}}{(\dot M c_p)_{oil}} \quad . \qquad (25)$$

Subroutine FLASH

This subroutine models a flash valve, flash tank, and feedwater make-up valve (see Fig. 6). All enthalpies are calculated as the product of specific heat and temperature, with the exception of saturated steam enthalpy h_3, which is determined from a separate steam-properties subroutine (4).

Flash Valve. We assume that the pressure upstream of the flash valve is maintained at 5 psi greater than the saturation pressure (determined from the steam-properties subroutine) in order to prevent boiling. Saturation pressure depends on the temperature at that point. The ΔP across the flash valve is, then, the difference between that upstream pressure and the saturation pressure of steam at the desired (input) delivery temperature. Parasitic power is calculated using this ΔP. If the upstream temperature is insufficient to supply steam at the desired temperature, the flash valve ΔP is arbitrarily set at 5 psi.

Flash Tank. First, the flash tank temperature is calculated as a function of time, with the same algorithm as that used in the storage tank subroutine. We assume for this calculation that no steam is delivered. Since the flash valve is a con-

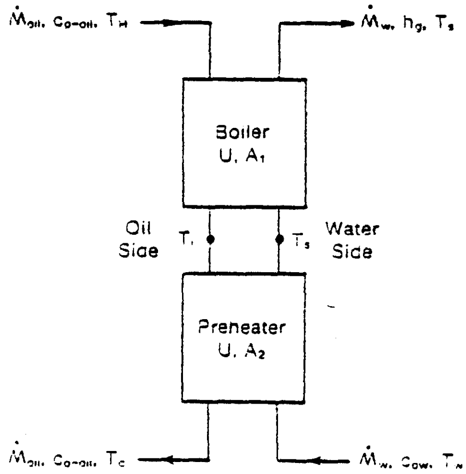

Fig. 5 SOLIPH Unfired-Boiler Model

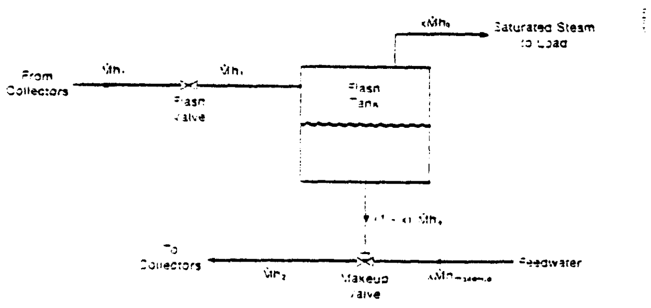

Fig. 6 SOLIPH Flash System Components and Energy Values

stant enthalpy device, this calculation is performed as if the upstream water were directly entering the flash tank. Thus,

$$T = T_\infty + (T_o - T_\infty)\, e^{-\frac{\dot{M}_h c_p + UA}{Mc_p} t} \quad , \qquad (26)$$

where

$$T_\infty = \frac{\dot{M}_h c_p T_h + UAT_{amb}}{\dot{M}_h c_p + UA} \quad , \qquad (27)$$

and $\dot{M}_h c_h$ is determined upstream of the flash valve. If the temperature T at the end of the time step is less than the steam-delivery temperature, no steam is delivered, and tank loss is calculated as it is in the storage subroutine:

$$\dot{Q}_{loss} = UA\,(T_\infty - T_{amb})\, t' + UA\,\frac{Mc_p}{X}\,(T_o - T_\infty)\left[1 - e^{-\frac{X}{Mc_p} t'}\right] , \qquad (28)$$

where $X = \dot{M}_h c_p + UA$. If the final calculated temperature exceeds the steam-delivery temperature, and the initial temperature was less than the steam temperature, we must determine how long it took to reach the delivery temperature, since heating above this point is not physically possible. To do this, we replace T with T_{steam} in the temperature equation and solve for time, t_{boil}. Energy loss during the heat-up period can be determined by substituting t_{boil} for t' in the energy loss equation. For time step t', the time left after boiling is $t_{left} = t' - t_{boil}$. During time t_{left}, the tank loss is simply

$$Q_{ss,loss} = UA(T_{steam} - T_{amb})t_{left} \quad . \qquad (29)$$

To determine the amount of steam delivered, the steam quality must be calculated from an energy balance:

Enthalpy in = steam enthalpy + recycled water enthalpy + heat losses + tank energy increase,

$$\dot{M}h_1 = x\dot{M}h_g + (1 - x)\dot{M}h_\ell + \dot{Q}_{loss} + \dot{Q}_{inc} \quad . \qquad (30)$$

Solving for quantity x,

$$x = \frac{\dot{M}h_1 - \dot{M}h_\ell - \dot{Q}_{loss} - \dot{Q}_{inc}}{\dot{M}(h_g - h_\ell)} \quad . \qquad (31)$$

The tank energy increase term is just

$$\dot{Q}_{inc} = Mc_p(T^t - T^{t-\Delta t}) \quad . \qquad (32)$$

Energy delivered to the load is the difference between enthalpy of delivered steam and enthalpy of the make-up water:

$$\dot{Q}_{del} = x\dot{M}h_g - x\dot{M}h_{make-up} \quad . \qquad (33)$$

Make-up Valve. One more energy balance is needed to determine the temperature of the water returned to the collectors. Referring to Fig. 6, we find

Energy to collectors = energy from the tank

+ energy of make-up water,

$$\dot{M}h_2 = (1 - x)\dot{M}h_\ell + x\dot{M}h_{make-up} \quad , \qquad (34)$$

or

$$h_2 = (1 - x)h_\ell + xh_{make-up} \quad . \qquad (35)$$

Table 2 Comparison of SOLIPH Energy Outputs With Other Computer Models Used in SSEA Sample IPH Problem

Computer Model	Open-Loop Low-Temperature Load					Closed-Loop High-Temperature Load	
	IHOR (MJ/m^2)	ICIN (GJ)	IACPT (GJ)	QCOUT (GJ)	QSOUT (GJ)	QCOUT (GJ)	QSOUT (GJ)
January							
DOE-2	345.0	441.0	419.0	171.0	164.0	96.0	73.0
TRNSYS (UW)	345.0	444.0	421.0	172.0	167.0	98.0	74.0
TRNSYS (ALTAS)	345.0	444.0	423.0	173.0	166.0	100.0	76.0
LASL	345.0	444.0	425.0	172.0	170.0	90.0	72.0
SOLIPH	344.6	443.4	424.4	172.9	166.8	98.4	77.4
July							
DOE-2	376.0	652.0	590.0	241.0	235.0	183.0	166.0
TRNSYS (UW)	375.0	645.0	584.0	241.0	235.0	179.0	163.0
TRNSYS (ALTAS)	375.0	645.0	588.0	242.0	234.0	184.0	167.0
LASL	375.0	647.0	593.0	243.0	239.0	174.0	163.0
SOLIPH	374.7	644.3	592.8	241.3	235.1	183.3	167.9
Yearly Total							
DOE-2	7625.0	6980.0	6408.0	2622.0	2472.0	1838.0	1559.0
TRNSYS (UW)	7625.0	6987.0	6396.0	2626.0	2485.0	1828.0	1552.0
TRNSYS (ALTAS)	7625.0	6987.0	6431.0	2646.0	2482.0	1880.0	1588.0
LASL	7628.0	6980.0	6468.0	2639.0	2531.0	1752.0	1545.0
SOLIPH	7624.7	6970.0	6492.0	2653.0	2501.0	1863.0	1595.0

The return temperature to the collectors T_2 is, then, h_2/c_p.

VERIFICATION OF SOLIPH

Because SOLIPH was written to make thousands of runs to generate design tools, it was important to check its accuracy. This was done using two methods: a "point" comparison of a specific detailed model with other recognized computer models and a parametric analysis compared with a recognized simplified method. Fortunately, an IPH system had already been modeled simultaneously with several other codes. In mid-1979, a Systems Simulation and Economic Analysis Working Group sponsored by DOE modeled a sample problem IPH system with TRNSYS (two different programmers), DOE-2, and a code written at Los Alamos Scientific Laboratory, which we will refer to as LASL. An attempt was also made to use the SOLTES program, but the user was unable to obtain satisfactory results in time. The example problem consisted of a CPC solar collector array, a heat exchanger, and a thermocline storage tank (5).

To check SOLIPH's performance, one of the task members modeled the sample problem with SOLIPH. Approximately 10 hours elapsed from the time work began until good results were output. Much of this time was spent incorporating a thermocline storage tank and mix valve, which SOLIPH did not have at that time. Table 2 shows monthly energy output results from SOLIPH compared with energy values supplied by the other programs. This comparison shows very good agreement for both the low- and high-temperature load cases.

Hourly plots of the storage tank temperature are given in Fig. 7. Comparisons between the codes of energy collected and energy delivered on an hourly basis are shown in Figs. 8 and 9. In all cases, the codes are shown to agree very well.

To test SOLIPH over a wide range of parameters, a variation study was conducted against FCHART 4.0. FCHART 4.0 is widely recognized and simple to run. A total of 20 annual runs of a simple flat-plate collector configuration were made with various combinations of the following 14 parameters:

- collector peak optical efficiency,
- collector U_L value,
- collector tilt,
- collector area,
- collector incident-angle modifier,
- ground reflectance,
- collector loop flow rate,
- steady-state pipe losses,
- storage size,
- storage skin losses,
- load flow rate,
- storage tank minimum discharge temperature,
- load profile, and
- load return temperature.

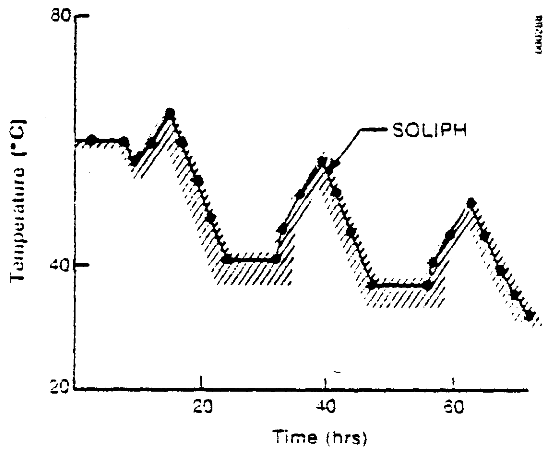

Fig. 7 **Bottom Storage Tank Temperature vs. Time for SOLIPH and SSEA Results**
(The shaded area represents the range of SSEA results.)

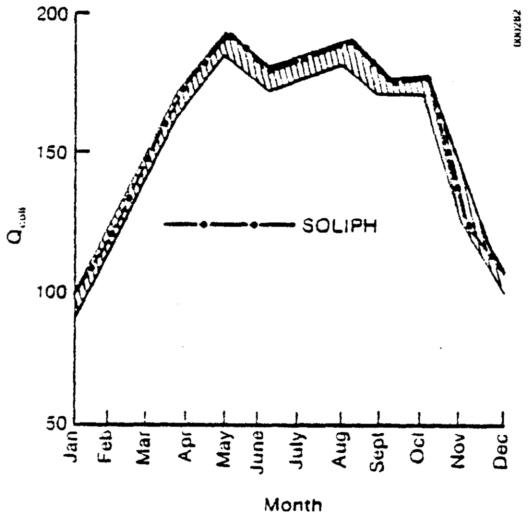

Fig. 8 **Energy Collection Comparison Between SOLIPH and SSEA Results**
(The shaded area represents the range of SSEA results.)

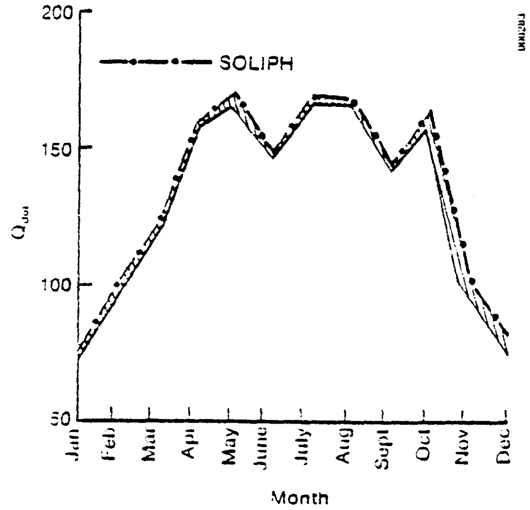

Fig. 9 **Energy Delivery Comparison of SOLIPH and SSEA Results**
(The shaded area represents the range of SSEA results.)

To permit a good comparison, all runs were made for one city, Albuquerque, and FCHART monthly insolation values were adjusted to agree with the TMY tape. Energy delivery values from SOLIPH and FCHART agreed to within 5% for 15 of the 20 runs. This agreement is considered good, since FCHART is a monthly code, and its accuracy is expected to be in the range of 5%. Five of the runs, however, showed differences of from 5% to 11%. In these runs, temperatures were higher than in the other cases. To determine the cause of these differences, we consulted with the University of Wisconsin Solar Laboratory, which developed FCHART and TRNSYS, the detailed hour-by-hour code from which FCHART was derived. Wisconsin Solar Laboratory made several TRNSYS runs for direct comparison with SOLIPH runs. TRNSYS and SOLIPH agree in each case to within 1-1/2%, but both exhibited differences from FCHART. The hypothesis was that the lower energy values from FCHART resulted because FCHART algorithms are more conservative at higher temperatures. In any case, the agreement between SOLIPH and TRNSYS further supported the accuracy of SOLIPH.

To check the accuracy of the SOLIPH TROUGH subroutine, parabolic trough runs were compared with Sandia-Albuquerque trough predictions. Again, excellent agreement was found. Thus, the basic nonsteam components of SOLIPH have been verified. However, the unfired-boiler and flash-tank subroutines, the most recently developed SOLIPH capabilities, have not yet been verified. Indeed, TRNSYS and FCHART do not have these capabilities. Recent comparisons of SOLIPH models with operating IPH field tests, however, lend credibility to their accuracy.

THE USE OF SOLIPH

Because of its lack of user-oriented special features, SOLIPH is relatively fast running. A year-long, hour-by-hour simulation of a system that includes a collector array, heat exchanger, storage, and associated piping takes about 10 CPU seconds of execution time. This amounts to a total cost of about $3.00 on SERI's CDC 7600 computer.

Since the original version of SOLIPH was completed in early 1981, IPH handbook task members have contributed a large number of improvements to the model. It is hoped that future funding might permit SOLIPH's capabilities to be exploited further, to generate results that could not be obtained before publication of this handbook.

ACKNOWLEDGMENTS

The author would like to gratefully acknowledge the contributions to the SOLIPH computer model made by the following people: Randy Gee (SERI) who wrote the parabolic trough subroutine and performed the verification versus F-chart; Kenneth May (SERI) who wrote the flash tank subroutine; Jim Leach (North Carolina State University) who, while working at SERI in the summer of 1981, wrote the first version of the unfired-boiler subroutine; Roger Davenport (SERI) who incorporated thermocline storage and mix valve capabilities and made the SSEA verification run; and Douglas Dougherty (SERI) who improved the piping subroutine.

REFERENCES

1. Kutscher, C. F. et al., "Design Approaches for Solar Industrial Process Heat Systems--Nontracking and Line Focus Collector Technologies," SERI/TR-253-1356, Aug. 1982, Solar Energy Research Institute, Golden, Colo.
2. Kutscher, C. F., "A Detailed Design Procedure for Solar Industrial Process Heat Systems: Overview, Proceedings of the ASME Solar Energy Division 6th Annual Technical Conference, SERI/TP-253-1830, 1982, Solar Energy Research Institute, Golden, Colo.
3. Gee, R., "A Simple Energy Calculation Method for Solar Industrial Process Heat Steam Systems," Proceedings of the ASME Solar Energy Division 6th Annual Technical Conference, 1982, Solar Energy Research Institute, Golden, Colo.
4. McClintock, R. B. and Silvestri, G. J., Formulations and Iterative Procedures for the Calculation of Properties of Steam, The American Society of Mechanical Engineers, New York, 1968.
5. Freeman, T.L., "A Comparison of the Predicted Performance of Several Solar System Simulation Codes for an Industrial Process Heating System," Proceedings of the Systems Simulation and Economic Analysis Conference, SERI/TP-351-431, Jan. 1980, Solar Energy Research Institute, Golden, Colo.